## CLAIMS

What is claimed is:

1. A computer system comprising:

a first section of non-volatile memory configured to store a BIOS program, the first section of non-volatile memory being reprogrammable; and

a second section of non-volatile memory operatively coupled to the first section of non-volatile memory, the second section of non-volatile memory being configured to store a boot-block program;

the boot-block program having a first validation routine configured to validate the BIOS program stored in the first section of non-volatile memory, and the BIOS program having a second validation routine configured to validate the boot-block program stored in the second section of non-volatile memory.

2. The system, as set forth in claim 1, wherein the first section of non-volatile memory comprises a protected segment of a reprogrammable memory device.

3. The system, as set forth in claim 2, wherein the second section of non-volatile memory comprises a reprogrammable segment of the reprogrammable memory device.

4. The system, as set forth in claim 3, wherein the reprogrammable memory device comprises a flash memory comprising the protected segment and the reprogrammable segment.

5. The system, as set forth in claim 1, wherein the first section of non-volatile memory comprises a first memory device.

6. The system, as set forth in claim 5, wherein the second section of non-volatile memory comprises a second memory device.

7. The system, as set forth in claim 1, wherein the boot-block program comprises a public key and a hash algorithm used to validate the BIOS program.

8. The system, as set forth in claim 7, wherein one of the boot-block program and the BIOS program comprises an encrypted hash correlative to the BIOS program.

9. The system, as set forth in claim 8, wherein the encrypted hash is encrypted using a private key correlative to the public key.

10. The system, as set forth in claim 9, wherein the boot-block program validates the BIOS program by calculating a first hash of the BIOS program using the hash algorithm, using the public key to decrypt the encrypted hash to produce a second hash, and comparing the first hash to the second hash.

11. The system, as set forth in claim 10, wherein the boot-block program does not allow the system to boot if the first hash does not match the second hash, and wherein the boot-block program does allow the system to boot if the first hash matches the second hash.

12. The system, as set forth in claim 10, wherein the system warns a user if the first hash does not match the second hash.

13. The system, as set forth in claim 12, wherein the boot-block program allows the system to boot if the first hash does not match the second hash.

14. The system, as set forth in claim 12, wherein the boot-block program allows the system to boot if the first hash does not match the second hash in response to an instruction to boot from the user.

15. The system, as set forth in claim 10, wherein various system resources are enabled or disabled depending upon whether the first hash matches the second hash.

16. The system, as set forth in claim 1, wherein the BIOS program comprises a public key and a hash algorithm used to validate the boot-block program.

17. The system, as set forth in claim 16, wherein one of the boot-block program and the BIOS program comprises an encrypted hash correlative to the boot-block program.

18. The system, as set forth in claim 17, wherein the encrypted hash is encrypted using a private key correlative to the public key.

19. The system, as set forth in claim 18, wherein the BIOS program validates the boot-block program by calculating a first hash of the boot-block program using the hash algorithm, using the public key to decrypt the encrypted hash to produce a second hash, and comparing the first hash to the second hash.

20. The system, as set forth in claim 19, wherein the BIOS program does not allow the system to boot if the first hash does not match the second hash, and wherein the BIOS program does allow the system to boot if the first hash matches the second hash.

21. The system, as set forth in claim 19, wherein the system warns a user if the first hash does not match the second hash.

22. The system, as set forth in claim 21, wherein the BIOS program allows the system to boot if the first hash does not match the second hash.

23. The system, as set forth in claim 21, wherein the BIOS program allows the system to boot if the first hash does not match the second hash in response to an instruction to boot from the user.

24. The system, as set forth in claim 19, wherein various system resources are enabled or disabled depending upon whether the first hash matches the second hash.

25. The system, as set forth in claim 1, comprising:

CMOS memory operatively coupled to at least one of the first section of non-volatile memory and the second section of non-volatile memory; and

non-volatile random access memory (NVRAM) operatively coupled to at least one of the first section of non-volatile memory and the second section of non-volatile memory.

26. The system, as set forth in claim 25, wherein the first validation routine is configured to validate at least one of the CMOS memory and the NVRAM.

27. The system, as set forth in claim 25, wherein the second validation routine is configured to validate at least one of the CMOS memory and the NVRAM.

28. The system, as set forth in claim 1, comprising a processing system operatively coupled to the first section of non-volatile memory and to the second section of non-volatile memory.

29. A computer system comprising:

means for validating a BIOS program stored in a first section of non-volatile memory; and

means for validating a boot-block program stored in a second section of non-volatile memory.

30. The system, as set forth in claim 29, wherein the means for validating the BIOS program comprises:

means for storing a public key and a hash algorithm used to validate the BIOS program;

means for storing an encrypted hash correlative to the BIOS program;

means for calculating a first hash of the BIOS program using the hash algorithm;

means for decrypting the encrypted hash using the public key to produce a second hash; and

means for comparing the first hash to the second hash.

31. The system, as set forth in claim 30, wherein the means for validating the BIOS program comprises:

means for booting the system if the first hash matches the second hash; and

means for not booting the system if the first hash does not match the second hash.

32. The system, as set forth in claim 30, wherein the means for validating the BIOS program comprises:

means for warning a user if the first hash does not match the second hash.

33. The system, as set forth in claim 32, wherein the means for validating the BIOS program comprises:

means for booting the system if the first hash does not match the second hash.

34. The system, as set forth in claim 32, wherein the means for validating the BIOS program comprises:

means for booting the system if the first hash does not match the second hash in response to an instruction to boot from the user.

35. The system, as set forth in claim 30, wherein the means for validating the BIOS program comprises:

means for enabling or disabling resources in dependence upon whether the first hash matches the second hash.

36. The system, as set forth in claim 29, wherein the means for validating the boot-block program comprises:

means for storing a public key and a hash algorithm used to validate the boot-block program;

means for storing an encrypted hash correlative to the boot-block program;

means for calculating a first hash of the boot-block program using the hash algorithm;

means for decrypting the encrypted hash using the public key to produce a second hash; and

means for comparing the first hash to the second hash.

37. The system, as set forth in claim 36, wherein the means for validating the boot-block program comprises:

means for booting the system if the first hash matches the second hash; and

means for not booting the system if the first hash does not match the second hash.

38. The system, as set forth in claim 36, wherein the means for validating the boot-block program comprises:

means for warning a user if the first hash does not match the second hash.

39. The system, as set forth in claim 38, wherein the means for validating the boot-block program comprises:

means for booting the system if the first hash does not match the second hash.

40. The system, as set forth in claim 38, wherein the means for validating the boot-block program comprises:

means for booting the system if the first hash does not match the second hash in response to an instruction to boot from the user.

41. The system, as set forth in claim 36, wherein the means for validating the boot-block program comprises:

means for enabling or disabling resources in dependence upon whether the first hash matches the second hash.

42. A method of operating a computer system comprising:

validating a BIOS program stored in a first section of non-volatile memory; and

validating a boot-block program stored in a second section of non-volatile memory.

43. The method, as set forth in claim 42, wherein the act of validating the BIOS program comprises:

storing a public key and a hash algorithm used to validate the BIOS program;

storing an encrypted hash correlative to the BIOS program;

calculating a first hash of the BIOS program using the hash algorithm;

decrypting the encrypted hash using the public key to produce a second hash; and

comparing the first hash to the second hash.

44. The method, as set forth in claim 43, wherein the act of validating the BIOS program comprises:

booting the system if the first hash matches the second hash; and

preventing the system from booting if the first hash does not match the second hash.

45. The method, as set forth in claim 43, wherein the act of validating the BIOS program comprises:

warning a user if the first hash does not match the second hash.

46. The method, as set forth in claim 45, wherein the act of validating the BIOS program comprises:

booting the system if the first hash does not match the second hash.

47. The method, as set forth in claim 45, wherein the act of validating the BIOS program comprises:

booting the system if the first hash does not match the second hash in response to an instruction to boot from the user.

48. The method, as set forth in claim 43, wherein the act of validating the BIOS program comprises:

enabling or disabling resources in dependence upon whether the first hash matches the second hash.

49. The method, as set forth in claim 42, wherein the act of validating the boot-block program comprises:

storing a public key and a hash algorithm used to validate the boot-block program;

storing an encrypted hash correlative to the boot-block program;

calculating a first hash of the boot-block program using the hash algorithm;

decrypting the encrypted hash using the public key to produce a second hash; and

comparing the first hash to the second hash.

50. The method, as set forth in claim 49, wherein the act of validating the boot-block program comprises:

booting the system if the first hash matches the second hash; and

preventing the system from booting if the first hash does not match the second hash.

51. The method, as set forth in claim 49, wherein the act of validating the boot-block program comprises:

warning a user if the first hash does not match the second hash.

52. The method, as set forth in claim 51, wherein the act of validating the boot-block program comprises:

booting the system if the first hash does not match the second hash.

53. The method, as set forth in claim 51, wherein the act of validating the boot-block program comprises:

booting the system if the first hash does not match the second hash in response to an instruction to boot from the user.

54. The method, as set forth in claim 49, wherein the act of validating the boot-block program comprises:

enabling or disabling resources in dependence upon whether the first hash matches the second hash.

55. The method, as set forth in claim 43, wherein the act of validating the BIOS program comprises:

performing at least one of a self-correcting, reset, and default function if the first hash does not match the second hash.